Везде вместо $O^*(f)$ пишем $O(f)$, опуская полиномиальную часть.
Decision/search problem:
есть ли x в массиве? найти позицию x в массиве. есть ли путь в графе? найти сам путь
Рассмотрим decision задачи
обозначим набор входов, на которых $true$ за L. Тогда надо проверить принадлежность.
P — работают за $O(C \cdot n^k)$.
NP — можно по подсказке проверить за полином, что лежит в L.
$P \subset NP$.
отсортирован ли массив? есть ли гамильтонов путь? $k - clique$
$coNP := \{L : \overline{L} \in NP\}$.
простое ли число?
$NP - hard \to$ к ним сводятся все NP задачи.
$NP - complete := NP - Hard \cap NP$
по определению сводятся все друг к другу.
$SAT, 3 - SAT, k - Clique, Ham, Set - Covering, k - independent \dots$

$$RP := \{L : \exists\, M \in P \begin{cases} x \notin L \Rightarrow M(x,y) = 0 \\ x \in L \Rightarrow Pr_y[M(x,y) = 1] \geqslant \frac{1}{2} \end{cases} \}$$
$$coRP := \{L : \exists\, M \in P \begin{cases} x \in L \Rightarrow M(x,y) = 1 \\ x \notin L \Rightarrow Pr_y[M(x,y) = 0] \geqslant \frac{1}{2} \end{cases} \}$$

То есть они ошибаются только в одну сторону. Так что повторив их кучу раз, они с очень экспоненциально растущей вероятностью выдадут правильный ответ.

Если алгоритм работает верно с вероятностью $p > 0$, то повторив его $\frac{1}{p}$ раз, мы получим константную вероятность, потому что $(1-p)^{\frac{1}{p}} \leqslant \frac{1}{e}$. Так что если повторить его $\frac{50}{p}$ раз, то он наверняка сработает.

ZPP — никогда не ошибающийся алгоритм, матожидание которого полиномиально.
Быстрая сортировка.
Поиск квадратичного вычета по модулю за $O(\log p)$.
Покраска графа в 3 цвета за 1.5^n.

Вычисление средней зарплаты.
Сведение $SAT \to 3 - SAT$.
Детерминированное решение $3 - SAT$ за 2^n, $3^{\frac{n}{2}} = 1.732^n$.
Рандомное за 1.5^n: $random - walk$ от случайного состояния на глубину n. $p \geqslant \sum\limits_k Pr[k]\frac{1}{3^k}$, где $Pr[k] = \frac{c(n,k)}{2^n}$ — вероятность, что расстояние равно k. Такая сумма равна на самом деле $\frac{(1+\frac{1}{3})^n}{2^n} = (\frac{2}{3})^n$ по биному Ньютона. Так что надо повторить $O(1.5^n)$ раз.
Если идти не n шагов, а $3n$, то вероятность $(\frac{3}{4})^n$ (без доказательства), так что время работы $O(1.333^n)$.
Как ходить? Ищем кафе. 3 варианта:
1. bfs
2. $random - walk$ без остановки
3. $random - walk$ на фиксированную глубину с возвратом.
Лас-Вегас и Монте-Карло.
$random - walk$ с оптимизациями хорошо работает в поиске Гамильтонова пути.
Гамильтонов цикл с полиномом памяти: формула включений-исключений.

$Random - shuffle$
$mt19937$ $rnd(239) \to$ на олимпиадах рандом не должен меняться при повторном запуске.
(можно $mt64$)
Найти мажорирующий элемент за $O(n)$.
Найти арифметическую прогрессию длины n в массиве длины $2 \cdot n$.

$0 - 1$-дерево
Игра выиграл первый/второй.
Дерево решений.
Случайный ход, если не повезло, в другое.
$L_n = 2 \cdot W_{n-1}$.

$W_n = \frac{1}{2}L_{n-1} + \frac{1}{2}(L_{n-1} + W_{n-1}) = L_{n-1} + \frac{1}{2} \cdot W_{n-1} = 2 \cdot W_{n-2} + \frac{1}{2} \cdot W_{n-1}$.
... решаем рекуренту.
$O(1.686^n)$.
$OR - AND$-дерево эквивалентно $0 - 1$ игре.
$min - max$-дерево.
Бинпоиск + $0 - 1$-дерево
$\alpha - \beta$ отсечение. Храним мин и макс выигрыш, если пойдем в другое место.
Есть оценка снизу на 1.414^n. На $0 - 1$-дереве есть оценка сверху на 1.686^n.